METHOD OF PART FLOW MODEL FOR PROGRAMMABLE LOGIC CONTROLLER LOGICAL VERIFICATION SYSTEM

CROSS-REFERENCE TO RELATED APPLICATION(S)

The present application claims the priority date of co-pending United States Provisional Patent Application Serial Number 60/236,964, filed September 29, 2000.

BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention relates generally to programmable logic controllers and, more specifically, to a method of part flow model for a programmable logic controller logical verification system for manufacturing a motor vehicle.

2. Description of the Related Art

It is known that programmable logic controller code is written by controls engineers after assembly tooling designs are completed and a manufacturing process has been defined. The creation of the programmable logic controller code is mostly a manual programming task with any automation of the code generation limited to "cutting and pasting" previously written blocks of code that were applied to similar

manufacturing tools. Once the programmable logic controller code is written, it is used by a programmable logic controller to build subsequent hard tools used in the manufacture of parts for motor vehicles. The programmable logic controller code is not validated (debugged) until the hard tools are built and tried. A significant portion of this tool tryout process is associated with the debugging of the programmable logic controller code at levels of detail from a tool by tool level, to a workcell level, and finally at a tooling line level.

It is also known that a manufacturing line is typically made of three to twenty linked workcells. Each workcell consists of a fixture to position product (sheet metal) and associated automation (robots) that process the product (welding). The workcell typically consists of a fixture/tool surrounded by three or four robots. The product is then transferred to the next workcell in the manufacturing line for further processing, until it exits the manufacturing line.

It is further known that the workcells for a manufacturing line can be modeled before the manufacturing line is implemented. The modeling techniques, such as Robcad

from Tecnomatix and Igrip from Deneb, for the manufacturing process are limited in scope to a workcell level, due to how these type of technologies represent and manipulate three dimensional data and tool motions. It is still further known that there are two PLC simulation systems commercially available, one from SST called PICS and the other from CAPE Software called VPLink. However, neither simulation system has an explicit part flow model for discrete part manufacturing. It is further known that part flow model simulation software exists known as "Discrete Event Simulation". Although Discrete Event Simulation software uses an explicit part flow model, the parts are active and engage resources as they move through the system. The software also makes use of an event clock that is incompatible with the needs of the PLC emulation for continuous time.

In manufacturing plant floor operations, programmable logic controllers (PLCs) execute PLC code through sensing where parts are located within the tooling by using sensors/switches. The representations of parts and part flow are unique and necessary within a Virtual PLC (VPLC) to accurately portray the logical condition of the manufacturing process to the inputs of a PLC. Without part movement in the

VPLC that can be "sensed", the emulated PLC would be unable to exercise its code for even its basic function of automatic sequencing.

Therefore, it is desirable to provide a method for application of a part flow model as part of a programmable logic controller logical verification system. It is also desirable to provide a method for logical modeling and simulation of parts that hold information specific to manufacturing processes that the parts have been routed through. It is further desirable to provide a method of part flow as a component of the VPLC to allow additional uses beyond verifying PLC logic such as routing and quality testing. Therefore, there is a need in the art to provide a method that meets these desires.

SUMMARY OF THE INVENTION

Accordingly, the present invention is a method of part flow model for a programmable logic controller logical verification system. The method includes the steps of constructing a part flow model, determining whether the part flow model is acceptable, and using the part flow model to test PLC code to build a manufacturing line.

One advantage of the present invention is that a method is provided for application of a part flow model as part of a programmable logic controller logical verification system. Another advantage of the present invention is that the method allows an operator to determine whether the PLC control design being planned will work as intended, prior to physically building the tools/manufacturing line. Yet another advantage of the present invention is that the method lessens the dependency of vendor tool tryout (VTTO) to prove out if the PLC control design is robust. Still another advantage of the present invention is that the method directly shortens product development timing, supporting faster vehicle program launch timing. A further advantage of the present invention is that the method allows a manufacturing line integrator to have an accurate representation of a control strategy without the cost of either developing a separate simulation or waiting until the plant floor is operating to debug their system. Yet a further advantage of the present invention is that the method allows both the controls developer and the information integrator to use the same part flow model present in the VPLC, resulting in substantial cost and time savings.

Other features and advantages of the present invention will be readily appreciated, as the same becomes better understood, after reading the subsequent description taken in conjunction with the accompanying drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a diagrammatic view of a system, according to the present invention, for using a method of part flow model for a programmable logic controller logical verification system illustrated in operational relationship with an operator.

FIG. 2 is a diagrammatic view of a method, according to the present invention, for application of part flow model as part of the programmable logic controller logical verification system of FIG. 1.

DESCRIPTION OF THE PREFERRED EMBODIMENT(S)

Referring to the drawings and in particular FIG. 1, one embodiment of a system 10, according to the present invention, for application of part flow model as part of a PLC logical verification system 18 to be described is shown. In the present invention, a user 12 uses a computer 14 for the

part flow model. The computer 14 sends and receives information from fixture design system 16a and workcell design system 16b via an electronic link. The fixture design system 16a and workcell design system 16b provide engineering data for standard components, tools, fixture models, and robots to interact with the fixture models. The computer 14 sends and receives information with a PLC logical verification system 18 via an electronic link. The PLC logical verification system 18 verifies the PLC logic for a workcell of a tooling or manufacturing line. The computer 14 also sends and receives information with a part flow design 20 via an electronic link. The part flow design 20 sends and receives information with the PLC logical verification system 18 to verify the PLC code. Once the PLC code is analytically verified, it is exported by the computer 14 via an electronic link to at least one PLC 22. The PLC 22 is then used at physical tool build to produce or build a workcell (not shown), which is used in a tooling line (not shown) for the manufacture of parts (not shown) for a motor vehicle (not shown). It should be appreciated that the computer 14, electronic links, and PLC 22 are conventional and known in the art.

Referring to FIG. 2, a method, according to the present invention, for application of part flow model as part of the PLC logical verification system 18 is shown. In general, the user 12 identifies part locations, including movement between stationary locations, on a VPLC workspace of the computer 14 using a part location editor of the computer 14. Each location has the capability of having resources attached to it, including part location switches. The collection of part locations make up a directed graph that, coupled with a part generator, allows the user 12 to visually see the flow of parts through the PLC logical verification system 18 by change of color (indicating the presence of a part) at any of the part locations. It should be appreciated that, once a basic part flow model has been implemented, the method may be extended in more elaborate data movement schemes.

The method begins by writing a control model file for standard component, tool, and fixture models by the fixture design system 16a. For example, the fixture design system 16a will create a control model definition that describes how four clamps need to be sequenced along with a hard-tooled welder. It should be appreciated that the control

model is information that describes events, dependencies, and logical conditions that are used to drive modeling software to be described.

The method reads the control model file from the fixture design system 16a by a workcell design system 16b. The method writes to a control model file for workcell models, incorporating information from the control model file from the fixture design system 16a. For example, if a fixturing station has robots or flexible automation involved, the workcell design system 16b will import the fixture and processing data defined in the control model file from the fixture design system 16a, allowing the user to add robots to interact with the fixture and clamps. It should be appreciated that several control model files may be written by the workcell design system 16b.

The method includes writing a control model file for part flow by the part flow design system 20. For example, the part flow design system 20 will create a part flow model definition that describes how a part flows through a workcell such as moving from a bin into a fixture. It should be appreciated that the part flow model is information that

describes events, dependencies, and logical conditions that represent or simulate part flow through the workcell.

The method reads and manages all control model files required to model a tooling or manufacturing line by the PLC logical verification system 18. The PLC logical verification system 18 also has the ability to manage large amounts of computer aided drafting (CAD) data. The PLC logical verification system 18 is predominately a collector of data and a viewing tool, not a creator of data.

The method writes a control model file by the PLC logical verification system 18 to "logically link" the fixtures of the fixture design system 16a and the workcells of the workcell design system 16b and the part flow of the part flow design system 20 into a tooling or manufacturing line. The method plays a control model by the PLC logical verification system 18, which is driven by the control model described within the control model files.

Referring to FIG. 2, a method, according to the present invention, for part flow model for the PLC logical verification system 18 is shown. The method starts with a part generator in bubble 100. The part generator is a representation of some part such as a vehicle quarter panel

selected by the user 12. The part generator generates a part type such as a front driver side quarter panel and serial number such as body style. From bubble 100, the method advances to block 102 and the part generator generates a unique part from a part location such as a bin of the parts. The method advances to bubble 104 and moves the part from the bin to another part location. To move the part, the user 12 selects a resource such as a robot having at least one capability such as a robotic arm in block 106 to move the part from the bin to another part location such as a fixture. The method advances to block 108 and loads the part in the fixture. The user 12 selects a fixture such as a clamp in block 109 to secure the part in the fixture. It should be appreciated that the part locations (moving and stationary) have status (part present or no part present). It should also be appreciated that the part moves through the PLC logical verification system 18 via color change on the computer 14.

The method may also proceed from bubble 110 by selecting another part generator as previously described. The method advances to block 112 and the part generator generates a part from a bin in block 112 as previously described. The method advances to bubble 114 and moves the part from the bin

as previously described. The method advances to block 108 and loads the part in the fixture as previously described. It should be appreciated that the fixture is a location to which the part is moved. It should also be appreciated that multiple parts may be represented with the method.

From block 108, the method advances to bubble 110 and moves the part from the fixture to block 114 and tests the part. The user 12 tests the logic by forcing a state in the control logic to test all exception logic. For example, the method tests for status as to whether the part is present or not present. After block 114, the method then ends. It should be appreciated that part locations have exit conditions that are interlocked. It should also be appreciated that a record exists with each part generated and that the individual resources can contribute information to the part record (such as an action performed or another part being bound to it). It should further be appreciated that the unique part record can be tested as it traverses the workcell, which allows subsystem capabilities such as quality and routing to be exercised. It should yet further be appreciated that the method may incorporate unique serialized parts, part types, and part assemblies. It should still further be appreciated that the

method is an iterative process between design and simulation carried out on the computer 14 by the user 12.

After the part flow model is designed, the method includes playing the part flow model by the PLC logical verification system 18. For example, the user 12 plays the part flow model by the PLC logical verification system 18 on the computer 14. The method includes determining whether the part flow model is acceptable. For example, the user 12 determines whether the part has traversed the workcell successfully. If the part flow model is not acceptable, the method includes modifying the part flow model. The user 12 uses the iterative process to change resources and capabilities of the part record and runs or simulates the part flow model with the PLC logical verification system 18 until it is acceptable to the user 12. Once the part flow model is acceptable to the user 12, the method includes generating PLC code and using the PLC code to build a manufacturing line. It should be appreciated that the part flow model is similar to a floor plan and is the basis for the PLC code.

The present invention has been described in an illustrative manner. It is to be understood that the

terminology, which has been used, is intended to be in the nature of words of description rather than of limitation.

Many modifications and variations of the present invention are possible in light of the above teachings. Therefore, within the scope of the appended claims, the present invention may be practiced other than as specifically described.